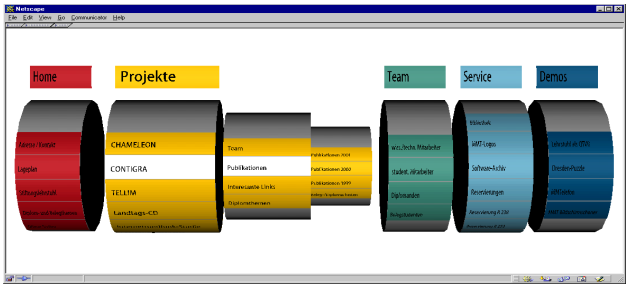
CONTIGRA: A High-Level Declarative Approach to Interactive 3D Components

Raimund Dachselt
Dresden University of Technology, Heinz-Nixdorf Endowed Chair for Multimedia Technology, dachselt@inf.tu-dresden.de

Motivation

This sketch reports on a new approach to facilitate the construction of interactive 3D graphics applications for the web and introduces the component architecture and underlying high-level languages. Considering the huge improvements in graphics hardware and the fast evolving internet technologies it seems surprising, that 3D applications are still not widely used. One problem is the dependency on proprietary 3D formats or the less successful VRML standard. Even with its promising successor X3D[1] the variety of web 3D formats will further exist. 3D graphics on the web is used in an increasing number of application areas such as product presentations, teaching or 3D navigation (example in figure 1). The second major problem is the lack of design standards, authoring tools and missing concepts of reuse. This is why building 3D applications is still time consuming and heavily depends on programming skills. However, interdisciplinary development of 3D solutions (e.g. for entertainment or shopping) using building blocks or 3D components is inevitably necessary.
The few existing component approaches like 3D-Beans[2] or i4D[3] have the disadvantage of 3D format dependency or usage of proprietary technologies. In addition to that the employed component technologies like JavaBeans are inherently code-centered and thus difficult-to-use for non-programmers. In this sketch we address these problems and propose a flexible solution.



The CONTIGRA Approach

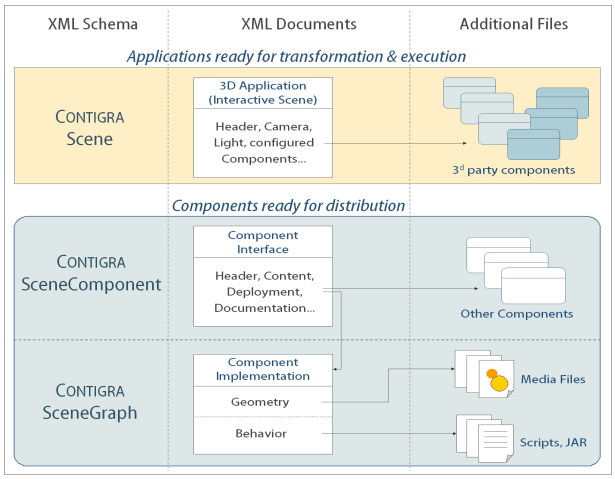
The acronym stands for Component OrieNted Three-dimensional Interactive GRaphical Applications. A 3D component concept is introduced that is largely independent of implementation issues and allows easy, declarative and interdisciplinary authoring of 3D applications. It is based on structured documents describing the component implementation, their interfaces and assembly / configuration. The core of the architecture are markup languages based on XML, allowing a consistent, declarative description of complex 3D scenes. XML has the advantages of hierarchical descriptions matching scene graph concepts, powerful transformation capabilities using DOM or XSLT, and interoperability with other web technologies. The 3 multi-layered CONTIGRA markup languages are coded with XML Schema.

CONTIGRA *SceneGraph*

As an extension to X3D this schema allows the implementation of a 3D component in terms of geometry and behavior, which are described separately. The set of scene graph nodes is extensible. Using XLST or DOM the documents can be translated to any 3D scene graph format. In addition to the *SceneGraph* component implementation there are resource files (sounds, textures, scripts…) all referenced in a homogeneous way (see figure 2).

CONTIGRA *SceneComponent*

This component description language is used to define component interfaces separated from their *SceneGraph* implementation, being well suited for distribution, search and deployment. Different sections of an interface document allow a rich component description from offered functionality and configurable parts up to deployment and authoring information. As an abstraction to implementation details high-level parameters hide scene graph fields. Other components can be encapsulated, pointers to *SceneGraph* documents and available editors are included.



CONTIGRA *Scene*

This is a high-level configuration language for component integration. Documents coded with this schema represent a declarative description of interactive 3D virtual environments. They contain a hierarchical assembly of configured component instances, component connections, and general scene parameters like cameras, lights etc.
How are these grammars applied? People with knowledge of scene graph concepts implement and describe 3D components using the *SceneGraph* and *SceneComponent* level. After distribution the components can be independently deployed and configured with an intuitive 3D user interface builder being under development. For this tool CONTIGRA *Scene* documents also serve as an exchange format. All the documents are finally transformed into a running 3D application, either at configuration time or on the fly in a web browser, adapted to the configuration of the client.
The declarative high-level languages and the runtime framework are still work in progress. Our preliminary results already show the flexibility and feasibility of the component-based approach to authoring 3D applications. Its major achievements are platform independence, abstraction to specific 3D formats, componentization and a declarative approach which is well suited for visual tool support.

References

1 Extensible 3D (X3D). http://www.web3d.org/x3d.html
2 R. Doerner and P. Grimm. Three-dimensional Beans – Creating Web Content Using 3D Components in a 3D Authoring Environment, Web3D/VRML 2000, February 2000.
3 C. Geiger, V. Paelke, C. Reimann, W. Rosenbach. A Framework for the Structured Design of VR/AR Content, VRST 2000, October 2000.